

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
27 February 2003 (27.02.2003)

PCT

(10) International Publication Number
WO 03/017619 A1

(51) International Patent Classification⁷: **H04L 29/06, 12/24**

(21) International Application Number: **PCT/NL01/00614**

(22) International Filing Date: 21 August 2001 (21.08.2001)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant (*for all designated States except US*): **TELEFONAKTIEBOLAGET L M ERICSSON [SE/SE]; S-126 25 Stockholm (SE).**

(72) Inventor; and

(75) Inventor/Applicant (*for US only*): **MOERDIJK, Ard-Jan [NL/NL]; Uitoord 42, NL-4824 LE Breda (NL).**

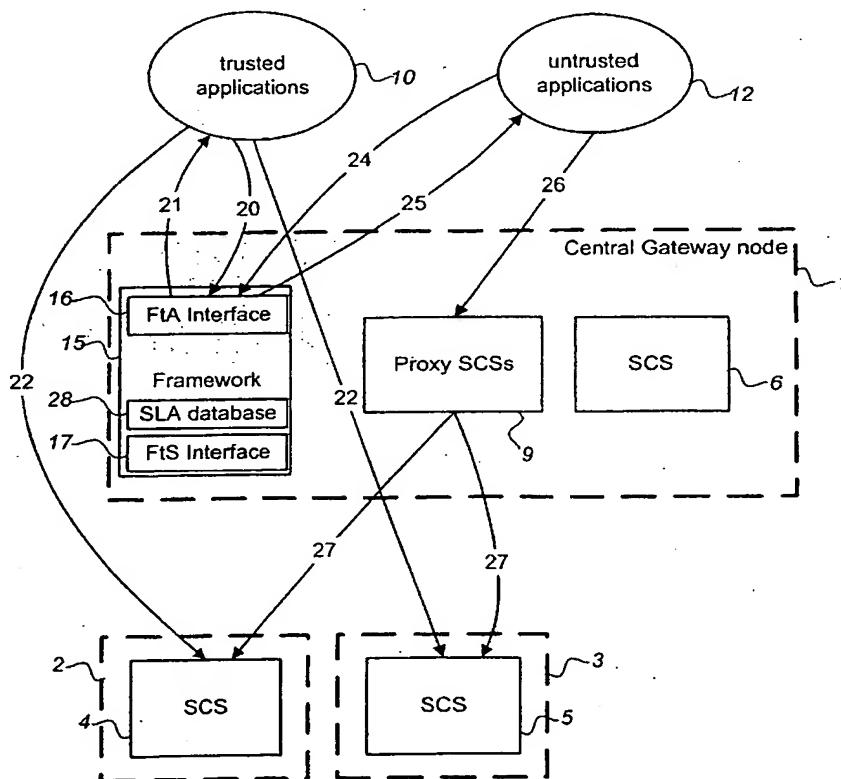
(74) Agents: **JORRITSMA, Ruurd et al.; Nederlandsch Octrooibureau, Scheveningseweg 82, P.O. Box 29720, NL-2502 LS The Hague (NL).**

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE; LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: A SECURE GATEWAY WITH PROXY SERVICE CAPABILITY SERVERS FOR SERVICE LEVEL AGREEMENT CHECKING



(57) Abstract: The present invention describes a gateway in an Open Service Access (OSA) network where Service Level Agreement (LSA) checks are performed by a Framework (15) on a Central Gateway node (1). A distinction is made between applications (10) that can be trusted, like applications provided by the same firm as the gateway, and other applications (12) that are not trusted for security reasons. Access request coming from the applications for accessing Service Capability Servers (SCSs) (4, 5) are checked by the framework (15). Now the trusted applications (10) can get direct access to the Service Capability Servers (4, 5), but the untrusted applications (12) are only allowed to access so-called proxy SCSs (9) on the Central Gateway node (1). The proxy SCSs (9) have the same interface as the SCSs (4, 5) running on the distant nodes (2, 3), and are downloaded by the Framework (15) from the distant SCS nodes (2, 3) to the Central Gateway node (1) during an initialization phase.

WO 03/017619 A1



Published:

- with international search report
- with amended claims

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

A secure Gateway with proxy Service Capability Servers for Service Level Agreement checking

5 Field of the invention

The present invention relates to Open Service Access (OSA) for the implementation of a third generation wireless phone system called Universal Mobile Telecommunications System (UMTS). The invention particularly relates to securing access to Service Capability Servers (SCSs).

Prior Art

Due to the 3rd generation network structure, applications within the UMTS Service Network can make use of core network functionality by means of open Application Program Interfaces (APIs). The open APIs, originally specified by the Parlay Group, are standardised within the 3rd Generation Partnership Project (3GPP) CNS working group and the ETSI SPAN 12 group.

In the OSA architecture, the logical entities that implement the open APIs are called Service Capability Servers (SCSs). Additionally infrastructure APIs are specified called the Framework. Among other things, the Framework provides an API for registering an SCS, see Technical Specification Group Core Network, Open Service Access, Application Programming Interface, Part 3: Framework (Release 4), 3GPP Techn. Spec. 29.198-3 V4.1.0 (2001-06). See also ETSI ES 201 915-3 V0.0.4 (2001-06) Open Service Access; Application Programming Interface; Part 3: Framework.

Several ways of implementing the APIs in a network are thinkable. One way is to provide all API implementations within one physical network node (i.e., both the OSA Framework and SCS in the same node having protocols/interfaces to the various core network entities). This could be called a physical or single OSA gateway. A second way is a distributed approach. In this case, one node comprises central infrastructure software (Framework) and maybe a few SCS components, but the rest of the SCSs run

on different nodes. Now, the OSA gateway is a logical gateway and API implementations can run on distributed nodes (i.e., SCSs). This means that the different entities provide their own APIs. This approach could be called logical or distributed OSA gateway.

5

Security in OSA is very important because applications can also be provided by other business domains than the network operator who provides the network capabilities. In practice, it might be that the network operator puts restrictions on the functionality of a certain SCS to be used by an application. What is allowed or not allowed, is defined by 10 so-called Service Level Agreements (SLAs). A Service Level Agreement may consist of an off-line (e.g. by physically exchanging documents) and an on-line part. An application has to sign the on-line part of a service agreement before it is allowed to access any network service capability feature.

15 The physical or single node gateway approach has disadvantages from a performance point of view as all the communication has to go via the OSA gateway node. The distributed approach has disadvantages from a security point of view because applications need to get direct access to the network nodes. A big advantage of the latter approach is that no implementation of the SCS software at the central gateway 20 node is necessary for introducing new SCSs. In case of a true physical gateway solution, of course for each new SCS new software has to be implemented on the gateway.

Summary of the invention

25

The problem to be solved by the present invention is to provide a system and method that overcome the disadvantages of the prior art approaches.

The present invention relates to a gateway node for implementing a gateway in an open service access network between one or more applications and one or more external 30 service capability servers on service capability server nodes, comprising a framework, characterised in that said framework performs security checks on requests from applications to get access to one or more of the external service capability servers using service level agreements, stored in a database.

In the present invention the physical and logical gateway approach are combined to benefit from both prior art approaches.

Furthermore, the present invention relates to a gateway node as described above,
5 wherein said framework classifies the applications in trusted and untrusted applications.

Moreover, the present invention relates to a gateway node as described above, wherein
said Framework allows a trusted application direct access to said service capability
servers but allows a untrusted application only access to proxy service capability
10 servers running on the gateway node.

Also, the present invention relates to a gateway node as described above, wherein said
framework initiates downloading of application program interface functionality from
said service capability servers to said gateway node forming said proxy service
15 capability servers.

Furthermore, the present invention relates to a method for implementing a gateway
node as described above, comprising the steps of:

- (a) receiving a request from an application to access a
20 service capability server external to said gateway node,
characterised in that the method also comprises the steps of:
 - (b) downloading of application program interface functionality from said
service capability servers to the gateway node forming a proxy service
capability servers;
 - (c) classifying the application into either a trusted or an untrusted application;
 - (d) requesting said external service capability server to create an object
instance implementing required application program interface functionality;
 - (e) sending a reference to said object instance to said application if it is a
trusted application;
 - (f) sending a reference to said proxy SCS to said application if it is an
untrusted application;

(g) operating said proxy service capability servers to enable controlled communications between said untrusted application and said service capability server.

5 In the present invention on the fly software can be provided to a node in the network that can deal with the checking of the Service Level Agreement. This keeps the actual SCS from doing these checks ad therefore trusted applications are not bothered with the SLA checking overhead. Meanwhile untrusted applications do not have direct access to the actual SCSs and have to go through one specific node, which is beneficial from a
10 security point of view.

Brief description of the drawings

Below, the invention will be explained with reference to some drawings, which are
15 intended for illustration purposes only and not to limit the scope of protection as defined in the accompanying claims.

Fig. 1 shows a network system with a logical OSA Gateway with Proxy SCSs.

Fig. 2 shows a possible arrangement for a network node.

Fig. 3 shows a part of a network with two OSA Gateways.

20

Description of preferred embodiments

In figure 1, an example of a network system with a logical (distributed) OSA Gateway is shown. The logical OSA Gateway shown, contains a Central Gateway node 1, and
25 two SCS nodes 2, 3. In Figure 1 the physical nodes are depicted with dashed lines. On each of the SCS nodes 2, 3 one or more external SCSs 4, 5 are running. An internal SCS 6 may be present on the Central Gateway node 1 itself.

One of the main drivers behind OSA is opening of the networks also for parties outside the domain of the network operator (owning the SCSs and the rest of the core network).

30 Applications provided within the same domain as the network operator are usually addressed as trusted applications, while applications provided by enterprises of a different domain (so-called 3rd parties) are usually addressed to as untrusted

applications. That is why in figure 1 a distinction is made between trusted applications 10 and untrusted applications 12. The applications 10, 12 are depicted as elipses.

On Central Gateway node 1, a Framework 15 is able to communicate with the applications 10, 12 through Framework to Application (FtA) Interface 16. On the other hand, Framework 15 is communicating with the SCSs 4, 5, 6 through Framework to Service (FtS) Interface 17. One of the functions of the FtS Interface 17 is to register all the SCSs 4, 5, 6, so that they can be discovered by the applications 10, 12, using the FtA Interface 16.

10 In the present invention, the FtS-Interface 17 is extended by proxy initialisation software to make it possible to download one or more Proxy SCSs 9 from all the external SCSs 4, 5, to the Central Gateway node 1. The Framework 15 has access to a database with Service Level Agreement information, referred to as SLA database 28. In figure 1, the SLA database 28 is situated in the Framework 15. However it should be understood that the SLA database 28 can also be situated outside the Framework 15, and can even be situated on another network node. Everytime an application 10, 12 is requesting access to one of the external SCSs 4, 5, Framework 15 will look in the SLA database 28 for SLA information. If the trusted applications 10 want to access the external SCSs 4, 5, they first have to request the Framework 15 for a reference to the external SCSs 4, 5 see arrow 20. Then the Framework 15 checks to which extent the trusted applications 10 are allowed to use these external SCSs 4, 5, using the information in the SLA database 28. If access is allowed, the Framework 15 requests the SCSs 4, 5 to create an object instance, implementing the requested APIs, on the SCS nodes 2, 3. Next, the Framework 15 returns references to these objects to the trusted applications 10, see arrow 21. The trusted applications 10 can then access the SCSs 4, 5 on SCS nodes 2, 3 directly, see arrow 22.

If untrusted applications 12 want to access the SCSs 4, 5, they also have to request the Framework 15 for a reference to the different SCSs, see arrow 24. The Framework 15 also checks to which extent the untrusted applications 12 are allowed to use these SCSs 4, 5, using the information in the SLA database 28. In reply to that, the Framework 15 creates SCS proxies 9 for the different SCSs 4, 5 on the Central gateway node 1. Then, as in the case of the trusted application, the Framework 15 requests the external SCSs

4, 5 to create an object instance implementing the requested APIs, on the SCS nodes 2, 3. Next the Framework 15 returns references to the Proxy SCSs 9 to the untrusted applications 12, see arrow 25. In this way, the untrusted applications 12 only have access to the Proxy SCS 9, see arrow 26, and the Proxy SCS 9 only invoke the implementations at the distant SCS nodes 2, 3 in case the untrusted applications 12 are allowed the invocation according to the SLA, see arrow 27. This means that all untrusted applications 12 are handled by the (Firewall of the) Central gateway node 1 and do not communicate directly with the SCS nodes 2, 3. This is beneficial from a security point of view.

10 There are multiple ways on how to provide the Proxy SCSs 9 at the Central gateway node 1. One solution is that each time a Service instance for a particular application needs to be initiated, the Framework 15 contacts the distant SCSs 4, 5 by invoking a method (for example the `getServiceManager()` on the `IpSvcFactory` interface, see 3GPP Technical Specification 29.198) with an indication that a Proxy SCS 9 is needed at the central Gateway node 1. The SCSs 4, 5 then initiate the proxy Service software with the correct SLA checks and return this to the Framework 15 together with a reference to the Service instance that is created locally on the SCS node 2, 3. The reference to the Service instance will link the Proxy SCS 9 at the Central Gateway node 1 to the Service instance at the SCS node 2, 3.

15 Another solution is that during installation time the SCSs 4, 5 register a Proxy SCS 9 at the central Gateway node 1. Each time a Service instance is needed, the Framework 15 first contacts the Proxy SCS 9 by calling e.g. the `getServiceManager()` method on the `IPSvcFactory`, see 3GPP Technical Specification 29.198. The Proxy SCS 9 then contacts the distant SCS 4, 5 and makes sure the correct Service proxy at the Central gateway node 1 and Service instance at the SCS node 2, 3 is started. In a preferred embodiment, the new operation for registering a Proxy SCS 9 in the FtS Interface 17, could be named `registerProxy()` and is may be part of the `IpFwServiceRegistration API`, see 3GPP Technical Specification 29.198. A reference between the Proxy SCS 9 and the distant SCS 4, 5 is established by extending the current operation

20 `getServiceManager()` on an interface, called the Service Factory API, with a parameter specifying a reference to the Proxy SCS 9. The downloading of the proxy SCS code is achieved by means of e.g. Java object serialisation. It is basically the same mechanism as downloading an applet in a client webbrowser.

Figure 2 shows a schematic block diagram of possible arrangement for a network node, like the Central Gateway node 1 and the SCS nodes 2, 3, comprising processor means 41 with peripherals. The processor means 41 is connected to memory units 38, 42, 43, 44 which store instructions and data, one or more reading units 45 (to read, e.g., floppy disks 39, CD ROM's 40, DVD's, etc.), a keyboard 46 and a mouse 47 as input devices, and as output devices, a monitor 48 and a printer 49. For data-communication to other nodes, an interface card 50 is provided that is connected to a network 51. Other input devices, like a trackball and a touch screen, and output devices may be provided for.

10 The memory units shown comprise RAM 42, (E)EPROM 43, ROM 44 and hard disk 38. However, it should be understood that there may be provided more and/or other memory units known to persons skilled in the art. Moreover, one or more of them may be physically located remote from the processor means 41, if required. The processor means 41 are shown as one box, however, they may comprise several processing units 15 functioning in parallel or controlled by one main processor, that may be located remote from one another, as is known to persons skilled in the art.

It is observed that both trusted and untrusted applications 10, 12 may run on a similar arrangement as shown in Figure 2, including its alternatives as indicated above. Instead of physical I/O means 50, means for wireless communications (transmitters, receivers, 20 transceivers) may be provided for. Applications 10, 12 may run on mainframes, PC's, handheld computers, laptops, and mobile devices like mobile phones.

Figure 3 shows an embodiment of the present invention with two OSA Gateways. In figure 3 two Central Gateway nodes 1, 81 are shown. Both Gateway nodes 1, 81 can be accessed by trusted and untrusted applications at Application Servers of which only 25 two untrusted applications 12, 82 are shown. One of the logical OSA Gateways comprises Central Gateway node 1 and SCS node 2. A second logical OSA Gateway comprises Central Gateway node 81 and SCS node 2.

In the situation of figure 3, two untrusted applications 12, 82 request access to an external SCS 4 running on a SCS node 2, via two separate Central gateway nodes 1, 81.

30 Each Central Gateway node 1, 81 comprises a Framework 15, 85. The untrusted applications 12, 82 request the Frameworks 15, 85 for a reference to the external SCS 4. Now, Framework 15 will create a Proxy SCS 9 on the Central Gateway node 1, and Framework 85 will create a Proxy SCS 89 on the Central Gateway node 81.

Because in this example, both applications are untrusted, the Frameworks 15, 85 returns references to the Proxy SCSs 9, 89 to the untrusted applications 12, 82. The untrusted applications 12, 82 then only have access to the proxy SCSs 9, 89, see arrow 26 and 86 respectively. The Proxy SCSs 9, 89 will invoke implementations on the external SCS 4 (see arrows 27, 87) in accordance with SLAs stored at the corresponding Frameworks 15, 85.

In this case both applications are untrusted applications. It should be understood that several combinations of trusted and untrusted applications are possible, and that this example can be expanded by adding more applications and/or gateways and/or SCSs.

10

In the embodiments described above, the word "gateway" is used to describe a collection of APIs, which gives applications access to the core (telecom) network. In the embodiments a gateway comprises a Framework and one or more SCSs. However other configurations may be possible such as a gateway only comprising a Framework.

15 It should be understood that the definition of gateway as used in the embodiment does not limit the scope of the present invention.

List of abbreviations

API	Application Program Interface
OSA	Open Service Access/Architecture
5 SCS	Service Capability Server
SLA	Service Level Agreement
UMTS	Universal Mobile Telecommunications System

Claims

1. A gateway node (1) for implementing a gateway in an open service access network between one or more applications (10, 12; 82) and one or more external service capability servers (4, 5) on service capability server nodes (2, 3), comprising a framework (15),

characterised in that said framework (15) performs security checks on requests from applications (10, 12) to get access to one or more of the external service capability servers (4, 5) using service level agreements, stored in a database (28).

10

2. A gateway node (1) according to claim 1, wherein said framework (15) classifies the applications (10, 12) in trusted and untrusted applications.

3. A gateway node (1) according to claim 2, wherein said Framework (15) allows a trusted application direct access to said service capability servers (4,5) but allows a untrusted application (12) only access to proxy service capability servers (9) running on the gateway node (1).

4. A gateway node (1) according to claim 3, wherein said framework (15) initiates downloading of application program interface functionality from said service capability servers (3, 4, 5) to said gateway node (1) forming said proxy service capability servers (9).

5. Method for implementing a gateway node (1) as defined in claim 1, comprising the steps of:

- (a) receiving a request from an application (10, 12) to access a service capability server (4, 5) external to said gateway node (1),

characterised in that the method also comprises the steps of:

- (b) downloading of application program interface functionality from said service capability servers (4, 5) to the gateway node (1) forming proxy service capability servers (9);
- (c) classifying the application into either a trusted or an untrusted application;

11

- (d) requesting said external service capability server (4, 5) to create an object instance implementing required application program interface functionality;
- (e) sending (21) a reference (22) to said object instance to said application if it is a trusted application (10);
- 5 (f) sending (25) a reference (26) to said proxy service capability servers (9) to said application if it is an untrusted application (12);
- (g) operating said proxy service capability servers (9) to enable controlled communications between said untrusted application (12) and said service capability server (4, 5).

10

6. Method for implementing a gateway node (1), according to claim 5, wherein said proxy service capability servers (9) are formed on said gateway node (1) every time said one or more applications (10, 12; 82) need access to said service capability servers (4, 5).

15

7. Method for implementing a gateway node (1), according to claim 5, wherein said proxy service capability servers (9) are formed on said gateway node (1) during an installation time of said gateway node (1).

20

AMENDED CLAIMS

[received by the International Bureau on 13 December 2002 (13.12.02); original claims 3 and 5 amended; remaining claims unchanged (2 pages)]

1. A gateway node (1) for implementing a gateway in an open service access network between one or more applications (10, 12; 82) and one or more external service capability servers (4, 5) on service capability server nodes (2, 3), comprising a framework (15),
5 characterised in that said framework (15) performs security checks on requests from applications (10, 12) to get access to one or more of the external service capability servers (4, 5) using service level agreements, stored in a database (28).
10
2. A gateway node (1) according to claim 1, wherein said framework (15) classifies the applications (10, 12) in trusted and untrusted applications.
15
3. A gateway node (1) according to claim 2, wherein said framework (15) allows a trusted application direct access to said external service capability servers (4, 5) but allows a untrusted application (12) only access to proxy service capability servers (9) running on the gateway node (1).
20
4. A gateway node (1) according to claim 3, wherein said framework (15) initiates downloading of application program interface functionality from said service capability servers (3, 4, 5) to said gateway node (1) forming said proxy service capability servers (9).
25
5. Method for implementing a gateway node (1) as defined in claim 1, comprising the steps of:
 - (a) receiving requests from an application (10, 12; 82) to access one or more service capability servers (4, 5) external to said gateway node (1),
characterised in that the method also comprises the steps of:
 - (b) performing security checks on said requests using service level agreements, stored in a database (28);
30
 - (c) downloading of application program interface functionality from said service capability servers (4, 5) to the gateway node (1) forming proxy service capability servers (9);

- (d) classifying the application into either a trusted or an untrusted application;
- (e) requesting said external service capability server (4, 5) to create an object instance implementing required application program interface functionality;
- (f) sending (21) a reference (22) to said object instance to said application if it is
5 a trusted application (10);
- (g) sending (25) a reference (26) to said proxy service capability servers (9) to
said application if it is an untrusted application (12);
- (h) operating said proxy service capability servers (9) to enable controlled
communications between said untrusted application (12) and said service
10 capability server (4, 5).

6. Method for implementing a gateway node (1), according to claim 5, wherein said proxy service capability servers (9) are formed on said gateway node (1) every time said one or more applications (10, 12; 82) need access to said service capability servers
15 (4, 5).

7. Method for implementing a gateway node (1), according to claim 5, wherein said proxy service capability servers (9) are formed on said gateway node (1) during an installation time of said gateway node (1).

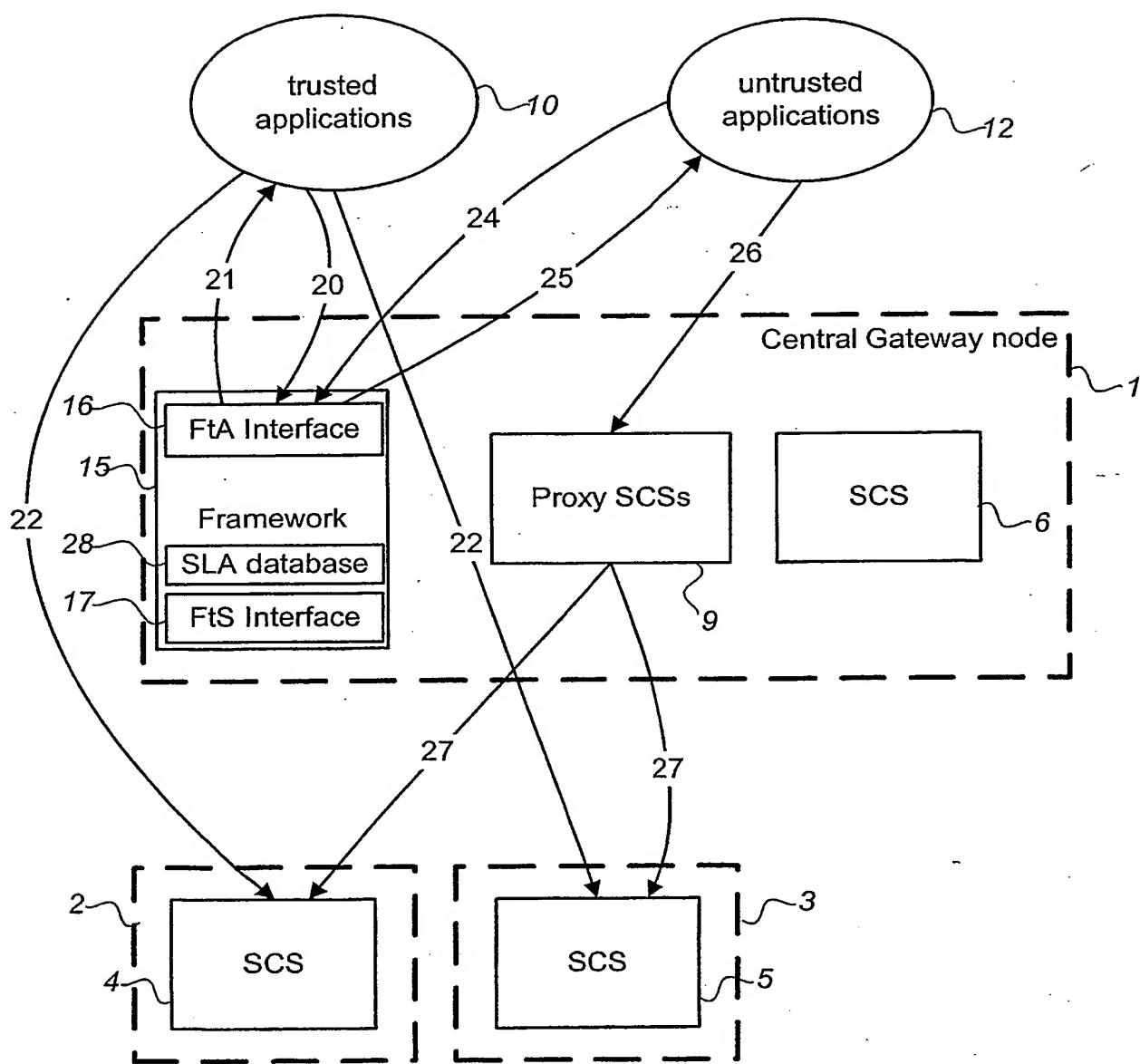
FIG. 1

FIG. 2

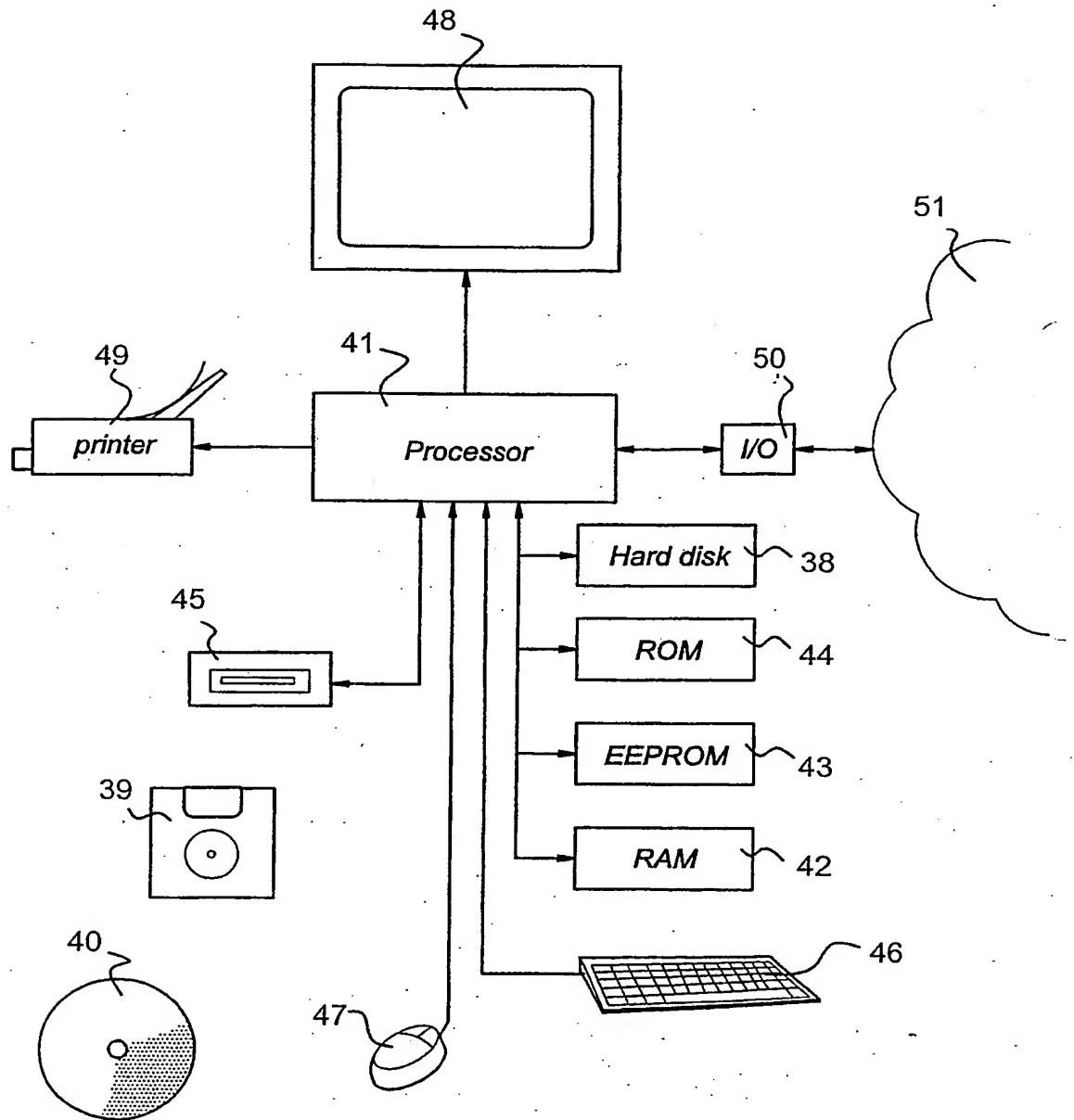
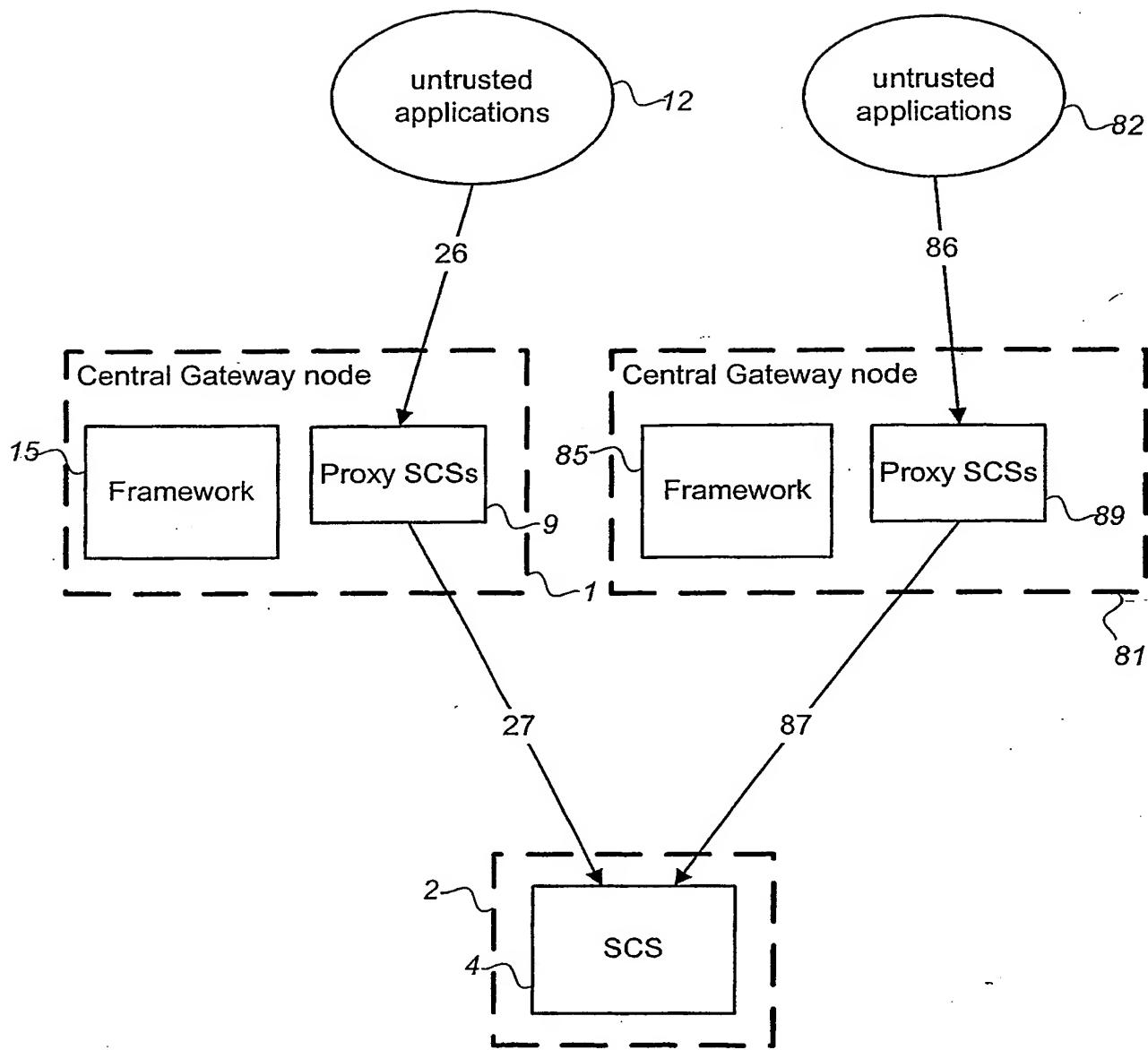


FIG. 3

INTERNATIONAL SEARCH REPORT

In Application No
PCT/NL 01/00614

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06 H04L12/24

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, IBM-TDB, COMPENDEX, INSPEC, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 00 79811 A (HUBER ADRIANO ; LAUPER ERIC (CH); SWISSCOM AG (CH)) 28 December 2000 (2000-12-28) page 1, line 18 -page 1, line 27 page 2, line 23 -page 3, line 2 page 3, line 33 -page 6, line 30 page 10, line 31 -page 11, line 11 figure 1	1-4
A	EP 1 107 550 A (CIT ALCATEL) 13 June 2001 (2001-06-13) column 1, line 1 -column 7, line 47 figure 1 abstract	1-4

Further documents are listed in the continuation of box C.

Patent family members are listed in annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

& document member of the same patent family

Date of the actual completion of the international search

16 July 2002

Date of mailing of the international search report

08.08.02

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Körbler, G

INTERNATIONAL SEARCH REPORT

Int'l Application No
PCT/NL 01/00614

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	BHOJ P ET AL: "SLA management in federated environments" COMPUTER NETWORKS, ELSEVIER SCIENCE PUBLISHERS B.V., AMSTERDAM, NL, vol. 35, no. 1, January 2001 (2001-01), pages 5-24, XP004304815 ISSN: 1389-1286 page 5, line 1 -page 11, line 25 page 13, line 1 -page 18, line 12 figures 1-7 abstract ---	1-4
A	EP 0 909 073 A (LUCENT TECHNOLOGIES INC) 14 April 1999 (1999-04-14) page 7, line 36 -page 8, line 28 figures 10A,10B abstract ---	1-4
A	STRETCH R: "The OSA API and other related issues" 1 January 2001 (2001-01-01) , BT , TECHNOL J VOL 19 NO 1 XP002206267 page 80, left-hand column, line 1 -page 85, left-hand column, line 44 figures 1-5 abstract ---	5-7
A	COOKSON M , SMITH D: "3G service control" 1 January 2001 (2001-01-01) , BT , TECHNOL J VOL 19 NO 1 XP002206268 page 72, right-hand column, line 16 -page 78, left-hand column, line 5 figures 4-7 abstract ---	5-7
A	BOS L, PETERS M: "An algorithm for automatically selecting the most suitable processing environment for mobile multimedia applications" 1 December 2000 (2000-12-01) , IEEE XP002206269 page 113, right-hand column, line 1 -page 117, right-hand column, line 16 figures 1-3 abstract -----	5-7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/NL 01/00614

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. Claims Nos.: because they relate to subject matter not required to be searched by this Authority, namely:

2. Claims Nos.: because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:

3. Claims Nos.: because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.

2. As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

The additional search fees were accompanied by the applicant's protest.

No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. Claims: 1-4

A gateway node for implementing a gateway between applications and service capability servers comprising a framework performing security checks on requests from applications by using service level agreements stored in a database.

2. Claims: 5-7

Method of receiving a request from an application to access a service capability server, downloading of application program interface functionality, classifying the application into either a trusted or an untrusted application, creating an object instance, sending a reference to the object and proxy service capability server.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/NL 01/00614

Patent document cited in search report	Publication date		Patent family member(s)		Publication date
WO 0079811	A 28-12-2000	WO AU EP	0079811 A1 4128799 A 1186185 A1		28-12-2000 09-01-2001 13-03-2002
EP 1107550	A 13-06-2001	EP AU CN JP	1107550 A1 7185000 A 1306380 A 2001249809 A		13-06-2001 07-06-2001 01-08-2001 14-09-2001
EP 0909073	A 14-04-1999	US EP JP	6098172 A 0909073 A2 11167537 A		01-08-2000 14-04-1999 22-06-1999

**ORIGINAL
NO MARGINALIA**